

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

----- x
:
UNITED STATES OF AMERICA
:
- v. -
:
JOSHUA ADAM SCHULTE,
:
Defendant.
:
----- x

DECLARATION
S3 17 Cr. 548 (JMF)

MICHAEL R. BERGER, pursuant to Title 28, United States Code, Section 1746, declares under penalty of perjury:

1. I am currently employed as a Computer Scientist with the Federal Bureau of Investigation ("FBI"), and have worked as a computer scientist for the FBI since September 2012. In that capacity, I am responsible for providing computer expertise to a wide range of FBI investigations, particularly with respect to the forensic analysis of digital evidence. A copy of my resume is attached as Exhibit A. In brief, I have Master of Science degrees in Computer Science and Forensic Computing, I am certified in a wide range of digital forensic analysis and cyber-incident response, and I teach a graduate-level course in Digital Forensics at the New York University Tandon School of Engineering.

2. In my capacity as an FBI computer scientist, I have provided assistance to the FBI special agents and Department of Justice prosecutors investigating this case. In February 2020, I testified as an expert witness in this matter. I make this affidavit based on my personal participation in this investigation and prosecution, as well as my training and experience in digital forensic analysis.

3. I have been provided with a copy of the affidavit of Steven M. Bellovin dated April 22, 2022 (the “Bellovin Affidavit”), and I make this affidavit to respond to certain claims made in the Bellovin Affidavit regarding my work in this matter and the ability of the defendant and his expert to review and dispute the conclusions that I made.

4. As an initial matter, I note that the Bellovin Affidavit’s description, in paragraph 6(b), of certain terminology related to digital forensics is inaccurate. A “forensic image” is not “forensic software.” Rather, a “forensic image” and a “mirror image” are terms that are used interchangeably. A “forensic case” is an entry in forensic software, such as the AccessData Forensic Toolkit (commonly referred to as “FTK”) referenced in the Bellovin Affidavit. A forensic case is simply a name assigned to the data that is imported into the forensic analysis software. A case can consist of a single forensic or mirror image, multiple forensic or mirror images, or other data that is imported into the software for analysis.

5. The Bellovin Affidavit also claims in paragraph 16 that “[n]o working programmer functions without full access to online documentation,” and makes similar claims in paragraph 37. While it is true that online reference material is a valuable resource in conducting digital forensic analysis, and I cannot speak to the particular experiences of the defendant’s expert in other matters (which do not appear to have involved digital forensics), experienced forensic examiners who are used to working with classified material frequently do not have concurrent access to Internet resources when analyzing classified information. I personally, in my participation in this case, on numerous occasions had to wait to conduct Internet research because Internet access was not readily available in the secure spaces where the highly-sensitive classified information in this case was maintained. There can be substantial security concerns about having unclassified, Internet-accessible computers present with computers that have sensitive classified information,

particularly without extensive safeguards on those unclassified computers. Many Secure Compartmented Information Facilities do not have any unclassified Internet access at all. Accordingly, while it may pose a challenge not to have concurrent access to classified information and Internet resources, it is a common occurrence for forensic analysts with experience in national security cases.

6. The Bellovin Affidavit claims, in paragraph 17, that I “relied upon all backups to note each day particular files were modified, and which of these versions were ultimately released by WikiLeaks,” and in paragraph 18, that I “could not have performed [my] timing analysis for the Stash or Confluence backups without a mirror image of the FS01 server, or at the very least, access to all the Stash and Confluence backup files in order to access and compare different versions of the backups to the leaked files.” That is inaccurate. My analysis was not performed with access to, or based upon, “all backups.” As my expert report and my testimony at trial made clear, my conclusions were based on two discrete forms of analysis applicable to Stash and Confluence that I also describe herein.

7. With regard to Stash, I, along with other FBI computer scientists, reconstituted the Stash database from the most recent available backup file into a working Stash environment. Stash is based on a system called “Git,” which records when changes to a file are “committed”—akin to when a file is saved—to a particular version of the file. The core components of Stash, like any Git system, are the “repositories” or “repos,” which contain the various versions of a particular file, and the “commit logs,” which record when particular changes were committed. By using these two components, a user is able to reconstitute any committed version of the file and identify the time at which that version came into existence.

8. Stash contained a variety of materials, including both source code and documentation, that were subsequently released by WikiLeaks. In order to conduct a comparison, I computed the “hash value,” which is a unique value generated by one of a variety of “hashing” algorithms, and which is specific to a particular file with particular content, for two of the source code files released by WikiLeaks, “Marble.horig” and “SolutionEvents.cs.” A “hash value” allows for comparison of two computer files. Files that are identical will have the same hash value when computed using the same algorithm, but even a change in a single bit between two files will result in noticeably different hash values. I then computed the hash values for six different versions of the “Marble.horig” source code file and five different versions of the “SolutionEvents.cs” source code file that appeared in the Stash repositories for those tools.

9. With respect to “Marble.horig,” I determined that the hash value for the source code file that appeared on WikiLeaks was the same as two versions of that file, which were last committed on February 26, 2016 at 9:36 a.m. and March 1, 2016 at 11:09 a.m. While it is impossible to be certain, the presence of two identical hash values in different versions of the same file may indicate that a user rolled back the data to the earlier version on March 1, 2016, and recommitted a previous version, in order to undo changes that had been made in the interim. The hash value for the source code that appeared on WikiLeaks was, however, different from a version of “Marble.horig” that was last committed on February 26, 2016 at 9:30 a.m. The fact that WikiLeaks had a version of “Marble.horig” saved on February 26, 2016 at 9:36 a.m., was the basis for my conclusion that the Stash data available on WikiLeaks originated no earlier than February 26, 2016 at 9:36 a.m.

10. With respect to “SolutionEvents.cs,” I determined that the hash value for the source code file that appeared on WikiLeaks was the same as the version of that file that was last

committed on February 13, 2016. I also determined that the hash value for the source code file that appeared on WikiLeaks was different from the version of "SolutionEvents.cs" that was last committed on March 4, 2016 at 9:45 a.m. The fact that this version did not appear on WikiLeaks was the basis for my conclusion that the Stash data available on WikiLeaks originated no later than March 4, 2016 at 9:45 a.m.

11. By combining my conclusions about Marble.horig and SolutionEvents.cs, I concluded that the Stash data available on WikiLeaks originated between February 26, 2016, at 9:36 a.m. and March 4, 2016 at 9:45 a.m.

12. I understand that the defendant's expert has been provided, on a standalone laptop, with the complete Stash repositories for all of the tools for which WikiLeaks released source code, along with the corresponding commit logs, derived from the most recent Stash backup available to the Government. I understand that the only redactions made to these files were to remove the usernames of the particular users who made commits, although the record of the commits themselves is complete. That data would permit a competent expert to review and test my conclusions about Stash, or to reach different conclusions. Stash data pertaining to tools that were not released by WikiLeaks would be of no value or relevance in evaluating my analysis, since there would be no point of comparison from WikiLeaks to determine whether a particular source code file with a particular hash value had been released or not. The data that was provided, however, would easily allow a competent expert to perform the same tasks that I did, by identifying the times of commits from the logs provided, hashing the files that correspond to those times, and saving those hash values for later comparison to the versions disclosed by WikiLeaks.

13. With respect to Confluence, my analysis was somewhat different. Because WikiLeaks modified the Confluence pages from their original form and posted them as PDF files,

hashing the content of those files would necessarily not yield any valuable comparisons—all of the files on WikiLeaks would have different hash values from the original versions that appeared on Confluence due to WikiLeaks' modifications of the files.

14. Instead, my analysis of the timing of the Confluence disclosures depending on the timing of different versions of particular Confluence pages. Every page revision in Confluence gets its own unique content ID, and as a user would update the page and new versions were created, Confluence would keep track of all the previous versions in the Confluence "SQL" database. SQL, or "Structured Query Language," refers to an extremely common programming language used to manage data held in a relational database, and it allows a user to conduct queries of the database to obtain data responsive to specific criteria. The Confluence pages released by WikiLeaks included lists showing the number of previous versions of the pages that were available. Reviewing the version history in the Confluence SQL database does not require access to "all" the Confluence backup files to "compare different versions of the backups to the leaked files." The SQL database from the backup of Confluence on any particular day includes the list of all versions of a particular page that had been saved as of that day.

15. I focused on two particular Confluence pages—"Michael R's home page" and "Build Felix LP." The version of "Michael R's home page" that appeared on WikiLeaks included, at the bottom, links to previous versions, displaying the numbers 1 through 16. Based on that, I concluded that WikiLeaks had released the seventeenth version of "Michael R's home page." Similarly, the version of "Build Felix LP" on WikiLeaks listed seven previous versions, indicating that WikiLeaks had released the eighth version of that page.

16. I loaded the SQL database for the most recent Confluence backup that was available to the Government, which was dated April 25, 2016, into a SQL review platform. Contrary to the

Bellovin Affidavit's assertions, I did not reconstitute the content files from that or any other backup—my analysis was based solely on the SQL database. Using standard, commonplace SQL queries, I generated a list of the available versions of “Michael R’s home page” and “Build Felix LP” and the times that those versions were created. Based on the lists generated in response to that query, I observed that the seventeenth version of “Michael R.’s home page”—that is, the version that was released on WikiLeaks—was saved on March 2, 2016 at 3:58 p.m. I also observed that while the eighth version of the Build Felix LP page—the version present on WikiLeaks—was created on March 2, 2016 at 8:01 a.m., there also existed a ninth version—one subsequent to the version disclosed by WikiLeaks—that had been created on March 3, 2016 at 6:47 a.m. By combining my analysis of the versions of these two files, I concluded that the data from Confluence disclosed by WikiLeaks came from between March 2, 2016 at 3:58 p.m., when the seventeenth version of “Michael R’s home page” that appeared on WikiLeaks was saved, and March 3, 2016 at 6:47 a.m., when the ninth version of Build Felix LP—which did not appear on WikiLeaks—was saved.


17. I then compared that data with a screenshot of the folder containing the daily backups of the Confluence database. Again, my comparison was based on the screenshot of the folder, not on the underlying backup files themselves. That screenshot showed the timestamps of when those individual backup files were created, typically a little before 6:30 a.m. each day. Only one backup file had been created during the window between March 2, 2016 at 3:58 p.m. and March 3, 2016 at 6:47 a.m.—the backup file that was created on March 3, 2016 at 6:29 a.m.

18. I understand that the defense in this case has been provided with, among other things, data from the Confluence backup files created on March 2, 3, and 4, and April 25, 2016. A competent forensic expert would be able to use that data to test my conclusion that the

Confluence material disclosed by WikiLeaks came from between March 2, 2016 at 3:58 p.m., and March 3, 2016 at 6:47 a.m. For example, using the SQL databases, the defendant's expert would be able to run the same basic SQL queries that I did for the version history of any page in the database, including the two pages on which I based my conclusions. Moreover, an expert—or even a basic computer user—could review the data to see either if there was content from the March 2, 2016 backup that did not appear on WikiLeaks, indicating that the source of the released data predated March 2, 2016; or conversely if there was content from the March 4, 2016 backup that did appear on WikiLeaks, indicating that the source of the released data post-dated that day. Because my conclusion was precise as to an approximately 16-hour window in which the released data could have originated, either the SQL database or the content files from the 24 hours on either side of that conclusion would readily be able to show if my conclusion was incorrect.

19. I declare under penalty of perjury that, to the best of my knowledge, the foregoing information is true and correct.

Dated: Nassau County, NY
April 29, 2022


Michael R. Berger
Computer Scientist
Federal Bureau of Investigation